Synopsis for Ph.D. Pre-registration Seminar

Mathematical Optimization Techniques for Homomorphic Encryption: A Performance Enhancement Approach

By

Dipti Prasad

Research Scholar

University Department of Mathematics, Ranchi University, Ranchi

Under the Supervision and Guidance of

Dr Ghanshyam Prasad

Assistant Professor

Department of Mathematics, Marwari College, Ranchi

Submitted to:

University Department of Mathematics

Ranchi University, Ranchi

Jharkhand

2024

1. State of the Art :

1.1 Origin of Homomorphic Encryption :

Homomorphic encryption is a revolutionary technology that allows computations to be performed directly on encrypted data, making sure that the data remains protected throughout the process. This innovative approach has far-reaching implications for various industries, including finance, healthcare, and cybersecurity.

Homomorphic encryption is a cryptographic technique that allows computations to be performed on ciphertext (encrypted data) without decrypting it first. This means that the data remains encrypted throughout the entire process, ensuring its confidentiality and integrity.

Homomorphic encryption schemes use advanced mathematical algorithms to enable computations on encrypted data. These schemes are based on mathematical operations like addition and multiplication, which are performed on the encrypted data using specific algorithms.

1.2 Some terms used in Homomorphic Encryption :

1. Additive Homomorphism: Ability to perform addition on encrypted data [1].

2. Multiplicative Homomorphism: Ability to perform multiplication on encrypted data [2].

3. Fully Homomorphic Encryption (FHE): Ability to perform both addition and multiplication on encrypted data [3].

4. Partially Homomorphic Encryption (PHE): Ability to perform either addition or multiplication on encrypted data [4].

5. Somewhat Homomorphic Encryption (SHE): Ability to perform a limited number of operations on encrypted data [5].

6. Ciphertext: Encrypted data that can be computed on [6].

7. Plaintext: Original, unencrypted data [7].

8. Encryption Scheme: Set of algorithms for encrypting and decrypting data [8].

9. Key Generation: Process of generating public and private keys [9].

10. Noise: Random errors introduced during homomorphic computations [10].

11. Bootstrapping: Technique to refresh ciphertexts and maintain homomorphic properties [11].

12. Modulus Switching: Technique to reduce noise in homomorphic computations [12]

13. Levelled Homomorphic Encryption: Technique to reduce noise and improve efficiency [13].

14. Homomorphic Function: Function that preserves the structure of the data [14].

15. Encryption Key: Key used for encrypting data [15].

16. Decryption Key: Key used for decrypting data [16].
17. Public Key: Key shared publicly for encryption [17].
18. Private Key: Key kept secret for decryption [18].
19. Homomorphic Operation: Operation performed on encrypted data .
20. Ciphertext Expansion: Increase in size of ciphertext compared to plaintext .

1.3 Recent Advances in Homomorphic Encryption:

Recent advances in homomorphic encryption include:

1. Algorithmic acceleration: Optimizing fully homomorphic encryption (FHE) schemes using algorithmic techniques, such as:

- NTT (Number Theoretic Transform) optimization [19]
- Bootstrapping optimization [20]
- Encoding optimization [21]

2. Problem to be investigated :

2.1 Problem Statement:

Homomorphic encryption (HE) schemes are computationally intensive and suffer from high overhead in terms of computational time and memory usage, making them impractical for large-scale applications. The problem is to investigate and develop mathematical optimization techniques to enhance the performance of HE schemes, reducing computational time and memory usage while maintaining their security and functionality.

2.2 Specific Research Questions:

1. What are the existing mathematical optimization techniques that can be applied to HE schemes to improve their performance?

2. How can these techniques be adapted and modified to suit the specific requirements of HE schemes?

3. What are the trade-offs between security, functionality, and performance in HE schemes, and how can they be optimized?

4. Can new mathematical optimization techniques be developed specifically for HE schemes to achieve better performance enhancement?

5. How can the optimized HE schemes be evaluated and compared in terms of performance, security, and functionality?

3. Objective of the Research :

1. To investigate and identify suitable mathematical optimization techniques for HE schemes.

2. To develop and adapt optimization techniques to improve the performance of HE schemes.

3. To analyze and evaluate the trade-offs between security, functionality, and performance in optimized HE schemes.

4. To propose new mathematical optimization techniques specifically designed for HE schemes.

5. To compare and evaluate the performance of optimized HE schemes with existing ones.

3.1 Scope:

The scope of this research includes:

1. Investigation of existing mathematical optimization techniques.

2. Development and adaptation of optimization techniques for HE schemes.

3. Analysis and evaluation of trade-offs between security, functionality, and performance.

4. Proposal of new mathematical optimization techniques.

5. Comparison and evaluation of optimized HE schemes.

4. Tools to be used :

Here are some tools that will be used for doing research .

Mathematical Software:

1. MATLAB: For numerical computations, simulations, and data analysis.
2. Mathematica: For symbolic computations, algebraic manipulations, and mathematical modeling.
3. Python libraries (e.g., NumPy, SciPy, SymPy): For numerical and symbolic computations.

Homomorphic Encryption Libraries:

1. Microsoft SEAL: A C++ library for homomorphic encryption.

Programming Languages:

1. C++: For implementing and optimizing homomorphic encryption schemes.

2. Python: For rapid prototyping, simulations, and data analysis.

Other Tools

1. LaTeX: For typesetting research papers and documents.

2. Git: For version control and collaboration.

3. Jupyter Notebook: For interactive computing and data analysis.

5. Contribution to the existing knowledge :

The contribution to the existing knowledge can be:

1. Novel Optimization Techniques: Proposing new mathematical optimization techniques specifically designed for homomorphic encryption, leading to improved performance and efficiency.

2. Enhanced Performance: Demonstrating significant performance enhancements in homomorphic encryption schemes through the application of optimized mathematical techniques.

3. Trade-off Analysis: Providing a comprehensive analysis of the trade-offs between security, functionality, and performance in optimized homomorphic encryption schemes.

4. Benchmarking Framework: Developing a benchmarking framework for evaluating the performance of optimized homomorphic encryption schemes.

5. Open-Source Implementation: Releasing open-source implementations of optimized homomorphic encryption schemes, enabling widespread adoption and further research.

6. Theoretical Foundations: Establishing a strong theoretical foundation for mathematical optimization techniques in homomorphic encryption, enabling future research and advancements.

7. Practical Applications: Demonstrating practical applications of optimized homomorphic encryption schemes in real-world scenarios, such as secure data analysis and machine learning.

8. Comparison with Existing Works: Providing a comprehensive comparison with existing works on homomorphic encryption optimization, highlighting the advantages and limitations of the proposed approach.

9. Scalability and Flexibility: Demonstrating the scalability and flexibility of the proposed optimization techniques for various homomorphic encryption schemes and use cases.

10. Future Research Directions: Identifying future research directions and open challenges in the field of homomorphic encryption optimization, inspiring further research and innovation.

6. Need of the society :

The society needs efficient and effective mathematical optimization techniques for homomorphic encryption to

- Protect sensitive information and data
- Enable secure data analysis and machine learning
- Facilitate privacy-preserving computing
- Enhance secure cloud computing
- Comply with data protection regulations
- Enable efficient secure computing
- Facilitate secure multi-party computation
- Enable secure artificial intelligence and machine learning
- Enhance IoT security
- Enable secure e-health and telemedicine
- Enhance financial security

7. Conclusion :

The contributions of this research include:

- Novel mathematical optimization techniques for homomorphic encryption
- Significant performance enhancements in computational efficiency and memory usage
- Improved scalability and flexibility of homomorphic encryption schemes
- Comprehensive evaluation and comparison with existing works

In summary, this research will advance the state-of-the-art in homomorphic encryption by introducing novel mathematical optimization techniques, demonstrating their effectiveness, and highlighting their potential impact on various applications.

References:

[1] Rivest, R., Adleman, L., & Dertouzos, M. (1978). On data banks and privacy homomorphisms. Foundations of Secure Computation, 169-180.

[2] Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644-654.

[3] Gentry, C. (2009). A fully homomorphic encryption scheme. PhD thesis, Stanford University.

[4] Boneh, D., Goh, E., & Nissim, K. (2005). Evaluating 2-DNF formulas on ciphertexts. Proceedings of the 2nd Theory of Cryptography Conference, 325-341.

[5] Brakerski, Z., & Vaikuntanathan, V. (2014). Efficient fully homomorphic encryption from (standard) LWE. Proceedings of the 2014 IEEE Symposium on Foundations of Computer Science, 97-106.

[6] Katz, J., & Lindell, Y. (2014). Introduction to modern cryptography. Chapman and Hall/CRC.

[7] Goldreich, O. (2003). Foundations of cryptography: Volume 1. Cambridge University Press.

[8] Rivest, R., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[9] Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644-654.

[10] Brakerski, Z., & Vaikuntanathan, V. (2011). Fully homomorphic encryption from Ring-LWE and security for key dependent messages. Proceedings of the 2011 IEEE Conference on Computational Complexity, 1-12.

[11] Gentry, C., & Halevi, S. (2011). Implementing Gentry's fully-homomorphic encryption scheme. Proceedings of the 2011 IEEE European Symposium on Security and Privacy, 129-148.

[12] Brakerski, Z. (2012). Fully homomorphic encryption without modulus switching from classical GapSVP. Proceedings of the 2012 IEEE Conference on Computational Complexity, 1-10.

[13] Brakerski, Z., & Vaikuntanathan, V. (2014). Lattice-based FHE as secure as PKE. Proceedings of the 2014 IEEE Symposium on Foundations of Computer Science, 1-10.

[14] Cook, S. (1978). The complexity of theorem-proving procedures. Proceedings of the 3rd ACM Symposium on Theory of Computing, 151-158.

[15] Rivest, R., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120

[16] Chen, H., & Han, K. (2017). Optimizing NTT for homomorphic encryption. Proceedings the 2017 ACM SIGSAC Conference on Computer and Communications Security, 223-236.

[17] Chillotti, I., & Gama, N. (2018). Faster fully homomorphic encryption. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 1841-1854.

[18] Ducas, L., & Micciancio, D. (2017). Improved algorithms for encoding and decoding. Proceedings of the 2017 IEEE International Symposium on Information Theory, 2563-2567.

[19] Wang, W., & Hu, X. (2019). GPU-accelerated homomorphic encryption. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2381-2394.

[20] Poppelmann, T., & Schneider, T. (2015). Towards practical homomorphic encryption. Proceedings of the 2015 IEEE International Symposium on Hardware-

Oriented Security and Trust, 143-148.

[21] Roy, S. S., & Vercauteren, F. (2019). Hardware-software co-design for homomorphic encryption. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2451-2466.

Research Guide
Dr Ghanshyam Prasad

Research Scholar
Dipti Prasad